Light field compression using translation-assisted view estimation

Baptiste Hériard-Dubreuil, Irene Viola, Touradj Ebrahimi
Multimedia Signal Processing Group (MMSPG)
Ecole Polytechnique Federale de Lausanne (EPFL)
Lausanne, Switzerland
firstName.lastName@epfl.ch

***Abstract*—Light field technology has recently been gaining traction in the research community. Several acquisition technologies have been demonstrated to properly capture light field information, and portable devices have been commercialized to the general public. However, new and efficient compression algorithms are needed to sensibly reduce the amount of data that needs to be stored and transmitted, while maintaining an adequate level of perceptual quality. In this paper, we propose a novel light field compression scheme that uses view estimation to recover the entire light field from a small subset of encoded views. Experimental results on a widely used light field dataset show that our method achieves good coding efficiency with average rate savings of 54.83% with respect to HEVC.**

***Index Terms*—light field compression, view estimation, light field coding**

I. Introduction

Light field photography has recently attracted the interest of the research community, as it allows to visualize and interact with three-dimensional scenes in a more realistic and immersive way. However, the increased volume of data generated in the acquisition requires new solutions to provide efficient storage and transmission. In particular, new compression solutions are needed to minimize the size of the light field data, while maintaining an acceptable visual quality.

Over the years, several solutions have been proposed to encode light field images. Some propose to exploit view synthesis or estimation to improve the coding efficiency. Jiang et al. use HEVC to encode a low-rank representation of the light field data, obtained by using homography-based low-rank approximation. They then reconstruct the entire light field by using weighting and homography parameters [1]. Zhao et al. propose a novel compression scheme that encodes and transmits only part of the views using HEVC, while the non-encoded views are estimated as a linear combination of the already transmitted views [2]. Viola et al. proposed a graph learning approach to estimate the disparities among the views, which can be used at the decoder side to reconstruct the 4D light field from a subset of views [3]. Astola et al. propose a method that combines warping at hierarchical levels with sparse prediction to reconstruct the 4D light field from a predefined set of perspective views [4], [5]. The solution was recently adopted as part of the JPEG Pleno Verification Model (VM) (WaSP configuration) [6]. Rizkallah et al. and Su et al. use CNN-based view synthesis to reconstruct the entire light field from 4 corner views, employing graph-based transforms [7] or 4D-shape-adaptive DCT [8] to encode the residuals. De Carvalho et at. propose the adoption of 4D DCT to obtain a compact representation of the light field structure [9]. The DCT coefficients are grouped using hexadeca-trees, for each bitplane, and encoded using an arithmetic encoder. The solution was also adopted as part of the JPEG Pleno VM (MuLE configuration) [6].

In this paper, we propose a new method that uses view estimation to reconstruct the 4D light field structure from a given subset of views, which are translated to account for the camera disparity among the views. To improve the reconstruction quality, residual encoding is implemented using Principal Component Analysis (PCA) to reduce the rate overhead. Results show that our method outperforms other state-of-the-art solutions in light field compression in terms of coding efficiency.

The paper is organized as follows. In Section II, we present in details the proposed approach. Section III illustrates the validating experiment, and in Section IV the results are presented and analyzed. Finally, we draw some conclusions in Section V.

II. Proposed approach

The global architecture of the encoder is represented in Figure 1. The encoder receives as parameter the 4D light field structure, along with the selected encoding parameters. Given the chosen subset of views to be encoded (reference views), it performs estimation of the remaining views. The reference views are compressed using HEVC/H.265 and transmitted to the decoder. Then, each view to be estimated is predicted through a linear combination of the compressed reference views, which are translated to account for the displacement among different views. The estimation is performed on a block basis, which are identified through quad-tree segmentation, to better account for the presence of several depth planes in the scene. The residuals for each estimated view are then computed, approximated using PCA, and transmitted to the

This work has been conducted in the framework of projects "Light field Image and Video coding and Evaluation" and "Advanced Visual Representation and Coding in Augmented and Virtual Reality" both funded by The Swiss National Foundation for Scientific Research under grant numbers 200021-159575 and 200021-178854.

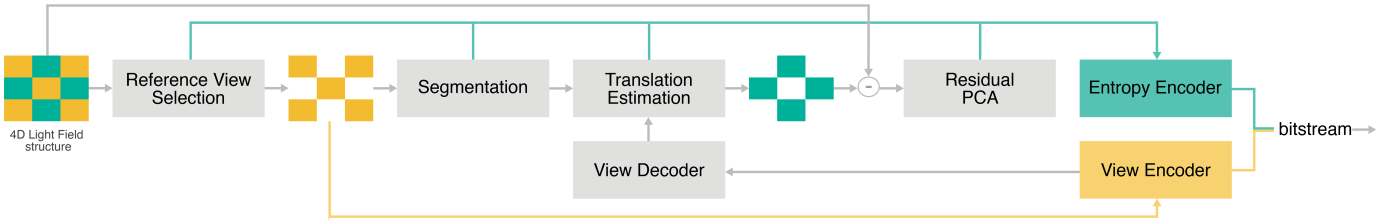


Fig. 1: Encoder architecture. The reference views are indicated in yellow, whereas the estimated views are indicated in green.

decoder along with the rest of the parameters for the view estimation.

At the decoder side, the reference views are decompressed, and the segmentation and prediction information is used to estimate the remaining views. The residuals are then added to obtain the final reconstructed views. In the following subsections, we will present in details the components of the encoder.

A. Segmentation

Given a point in a 3D scene $P = (V_x, V_y, V_z)$, the disparity between its projected points into two views A and B, p_A and p_B can be expressed as a function of the distance from P to the camera z (depth) and the translation between A and B t_{AB} [2]. Thus, in order to precisely estimate one point in a given view from a set of neighboring views, different translation factors should be assigned to each depth plane.

To limit the complexity of the encoder and the additional information to be sent to the receiver, we approximate the subdivision in different depth planes by using blocks, obtained by applying quad-tree segmentation. The segmentation is applied on the sum of the distance between the extreme horizontal views and the extreme vertical views of the Y channel, which will give us an estimation of the depth boundaries (this estimation will be used for all the views). More precisely, defining $I_{i,j}$ the Y channel of view (i, j), $i = 1, \ldots, M$ and $j = 1, \ldots, N$, with M and N representing the angular resolution of the 4D light field, and $|\cdot|$ denoting the absolute value operator, the sum of the distance C is computed as such:

$$C = \left|I_{\lfloor M/2 \rfloor,1} - I_{\lfloor M/2 \rfloor,N}\right| + \left|I_{1,\lfloor N/2 \rfloor} - I_{M,\lfloor N/2 \rfloor}\right| \quad (1)$$

.

As mentioned above, a quad-tree based algorithm is selected to compute the blocks. Through a configuration file, the user can decide the approximate number of blocks to be used in the estimation.

B. Reference compression

In order to achieve a good compression efficiency, the selected reference views are arranged into a pseudo-temporal video sequence and compressed using video codec HEVC/H.265 (version HM-15.0). However, it should be noted that any image or video compression can be used to encode the reference views. The position of the selected reference views is signalled using a binary mask, which is entropy encoded and sent to the decoder.

C. Translation estimation

For each view to be predicted, the predictor will express each block as the linear combination of a subset of blocks from the compressed reference views, after translation, to account for the camera disparity.

Using K reference views for the estimation, and defining as I_k the k-th reference view, $T_{i,j,k}$ the corresponding translated block i for view j with respect to view k, and $w_{i,j,k}$ the corresponding weight, we can compute the predicted block i of view j $\widetilde{I_j[i]}$ as follows:

$$\widetilde{I_j[i]} = \sum_{k=1}^{K} w_{i,j,k} \cdot I_k\left[T_{i,j,k}\right] \quad (2)$$

The translation parameters of $T_{i,j,k}$ are obtained using phase correlation:

$$\max_{x,y} \mathcal{F}^{-1}\left[\frac{\hat{I}_j \circ \hat{I}_k^*}{\left|\hat{I}_j \circ \hat{I}_k^*\right|}\right](x, y) \quad (3)$$

Where $\hat{I}_k$ and $\hat{I}_j$ are the reference and the translated views in the Fourier domain, $\mathcal{F}^{-1}$ is the inverse Fourier transform, $*$ is the complex conjugate, and $\circ$ is the Hadamard product.

The method works with subpixel precision; however, to limit the overhead, the translations were rounded to integers.

Considering all the translated references for object i in matrix X (each column is one translated reference), and defining Y as the ground truth, we then compute the linear combination weights solving the ridge regression problem, which is the minimization of the Mean Square Error (MSE) with a regularization term (squared 2-norm):

$$\min_W (Y - X \cdot W)^T (Y - X \cdot W) + \eta (W^T \cdot W), \quad (4)$$

Where W is the weight matrix, η is a regularization coefficient and $\mathbb{I}$ is the unity matrix. This formula actually admits an analytical solution:

$$W = \left(X^T \cdot X + \eta \mathbb{I}\right)^{-1} \cdot X^T \cdot Y \quad (5)$$

(a) *I01 (Bikes)* (b) *I02 (Danger_de_Mort)* (c) *I04 (Stone_Pillars_Outside)* (d) *I09 (Fountain_&_Vincent_2)*

Fig. 2: Central perspective view from each content used in the validating experiment.

This allows us to find the best coefficients in an efficient and robust way.

The weights are saved as 16-bit floating point numbers. To reduce the overhead in the total bitstream, only a subset of the reference views (neighbors) can be used to estimate each block.

D. Residuals

Residuals are computed between each original and estimated view. To exploit their redundancy in the angular plane, PCA is used to obtain the decomposition of the residuals, and only the first k coefficients are sent. This allows us to concentrate the variance of the residuals in the first coefficients, allowing for a significant quality improvement with a reasonably small number of coefficients.

III. Validating Experiment

In this section the validating experiment performed to test the performance of our solution is presented. Specifically, we outline the coding conditions in the codec configuration. We then briefly describe the anchors and, lastly, delineate how the objective metrics are computed.

A. Coding conditions

To allow for an easier comparison between the proposed approach and the state of the art in light field coding, the same conditions defined in the ICIP 2017 Grand Challenge were adopted for this experiment [10]. In particular, four light field contents were selected from the proposed lenslet dataset [11]: *I01 (Bikes)*, *I02 (Danger_de_Mort)*, *I04 (Stone_Pillars_Outside)* and *I09 (Fountain_&_Vincent_2)* (see Figure 2).

Each 10-bit lenslet image was devignetted and demosaicked; then, the Light Field toolbox v0.4 was employed to obtain the 4D light field structure of perspective views [12], [13]. A total of 15×15 perspective views were obtained from the lenslet image, each with a resolution of 625×434 pixels; however, only the central 13×13 views were selected to be encoded and evaluated. Color and gamma correction was applied to each perspective view prior to the encoding, following the JPEG Pleno Common Test Conditions [14].

The target bit rates are defined at 0.75 bpp (bits per pixel), 0.1 bpp, 0.02 bpp and 0.005 bpp. The bpp is computed as the total number of bits used for transmitting the encoded the picture divided by the number of pixel per channel ($13 \times 13 \times 434 \times 625$).

B. Encoding parameters

The references views were encoded in YUV420 format and 10 bit precision, using HEVC/H.265 reference encoder HM-15.0, with profile Main10 and low delay configuration. The segmentation blocks, translation coefficients and linear weights were computed on the luminance channel and entropy encoded. For the chroma channels, it was decided to use the same translation coefficient as the luminance channel, to reduce the overhead. Four references were used to perform the estimation, uniform weights were adopted for all the views, and no residuals were computed.

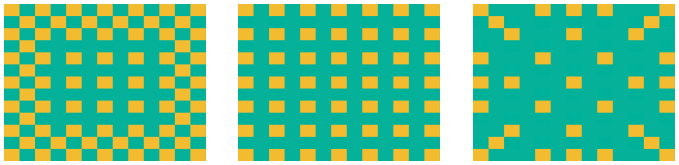
Table I summarizes the configuration parameters adopted for each light field content and for each target bit rate, while Figure 3 depicts the selected references for every configuration. Table II shows an example of how the bit rate was allocated, for content *Fountain_&_Vincent_2*.

TABLE I: Chosen parameters for the codec.

	bit rate (bpp)	**No. ref.**	**QP**	**No. neighbors**	**No. PCA components**	**No. blocks**
I01	0.75	69	13	69	10	~ 10
	0.1	49	22	49	0	~ 1
	0.02	49	30	16	0	~ 1
	0.005	33	39	8	0	~ 1
I02	0.75	69	13	69	10	~ 25
	0.1	49	23	49	0	~ 4
	0.02	33	31	33	0	~ 1
	0.005	33	40	4	0	~ 1
I04	0.75	69	13	69	10	~ 25
	0.1	49	21	49	0	~ 4
	0.02	33	26	33	0	~ 1
	0.005	33	33	4	0	~ 1
I09	0.75	69	13	69	10	~ 25
	0.1	69	23	40	0	~ 8
	0.02	49	30	16	0	~ 1
	0.005	33	39	4	0	~ 1

C. Anchor selection

To assess the performance of our coding approach, we compared it to the results obtained from HEVC/H.265 anchor



(a) 69 views (b) 49 views (c) 33 views

Fig. 3: Selected references (in yellow) for the codec configurations (see Table I).

TABLE II: Example of the bit rate repartition for each bit rate, in bpp (for content *I09*). The remaining bits carry the parameters.

Total bit rate	Reference views	Translations and weights	Residuals
0.8115	0.5689	0.0661	0.1763
0.1040	0.0967	0.0070	0
0.0189	0.0176	0.0009	0
0.0037	0.0031	0.0004	0

used in the ICIP 2017 Grand Challenge [10], using the software implementation x265[1].

In addition, our results were compared to the graph based solution proposed in [3]. This solution used 85 references encoded with the HEVC/H.265 reference software HM, and predicted the others using a graph learning approach. It yielded better results than the proposed solution of the 2017 ICIP Grand Challenge.

Finally, the JPEG Pleno VM was used as third anchor, using both the WaSP and MuLE configurations as they were provided in [6].

D. Objective quality metrics

To compare the results of our solution against the anchors, we used different objective quality metrics. Specifically, PSNR was computed on the Y and YUV channels, and SSIM was computed on the Y channel, following the JPEG Pleno Common Test Conditions [14].

IV. Results

Figure 4 depicts the performance of our proposed method with respect to the anchors in terms of $\widehat{PSNR}_Y$, $\widehat{PSNR}_{YUV}$ and $\widehat{SSIM}_Y$, for all target bit rates. Due to space constraints, results are shown only for content *I09*. Results show how our proposed solution outperforms the anchors across the bit rates. In particular, with metrics $\widehat{PSNR}_Y$ it performs similarly to the JPEG Pleno VM in the WaSP configuration for the highest bit rate, and to the MuLE configuration of the JPEG Pleno VM for $\widehat{PSNR}_{YUV}$. However, for lower bit rates our method is shown to be outperforming all the proposed anchors. Results of the computation of $\widehat{SSIM}_Y$ show that all codecs have similar performance for the highest bit rate, whereas for lower bit rates the superiority of our method is shown.

Results of the computation of Bjontegaard rate savings and PSNR gain are depicted in Tables III and IV, respectively. When compared to HEVC/H.265, our proposed method achieves on average a rating reduction of 58.46% and a PSNR gain of 2.57 dB for $\widehat{PSNR}_Y$ (54.83% rate reduction and 1.95 dB gain for $\widehat{PSNR}_{YUV}$). The maximum rate reduction is achieved for content *I04* (63.12% and 59.67% for $\widehat{PSNR}_Y$ and $\widehat{PSNR}_{YUV}$, respectively), whereas the biggest PSNR gain is obtained for content *I02* (2.72 dB and 2.13 dB for $\widehat{PSNR}_Y$ and $\widehat{PSNR}_{YUV}$).

Marginally higher rate reductions and PSNR gains are achieved when comparing our approach to the JPEG Pleno VM in its WaSP configuration, at least for $\widehat{PSNR}_Y$, as we reach an average reduction of 59.98% and an average PSNR gain of 2.70 dB. The best performance, in this case, is obtained for content *I01* (64.92% reduction and 3.11 dB gain). Our proposed method is also performing better than the JPEG Pleno VM with the MuLE configuration, as it achieves an average bit rate reduction of 49.93% for $\widehat{PSNR}_Y$ and 32.58% for $\widehat{PSNR}_{YUV}$ (respective average gains of 1.92 dB and 0.96 dB). The diminished performance when considering the chroma channels can be explained by the fact that for our approach, translation and linear combination weights were not computed for the chroma channels to reduce the overhead. Thus, lower performance can be expected with respect to the luminance channel.

Smaller, but still significant rate reduction and PSNR gain values can be observed with respect to the graph-based approach. In particular, our proposed solution achieves an average rate reduction of 24.18% and 22.27% for $\widehat{PSNR}_Y$ and $\widehat{PSNR}_{YUV}$, respectively, corresponding to a PSNR gain of 0.81 dB and 0.59 dB. The best performance is achieved for content *I09*.

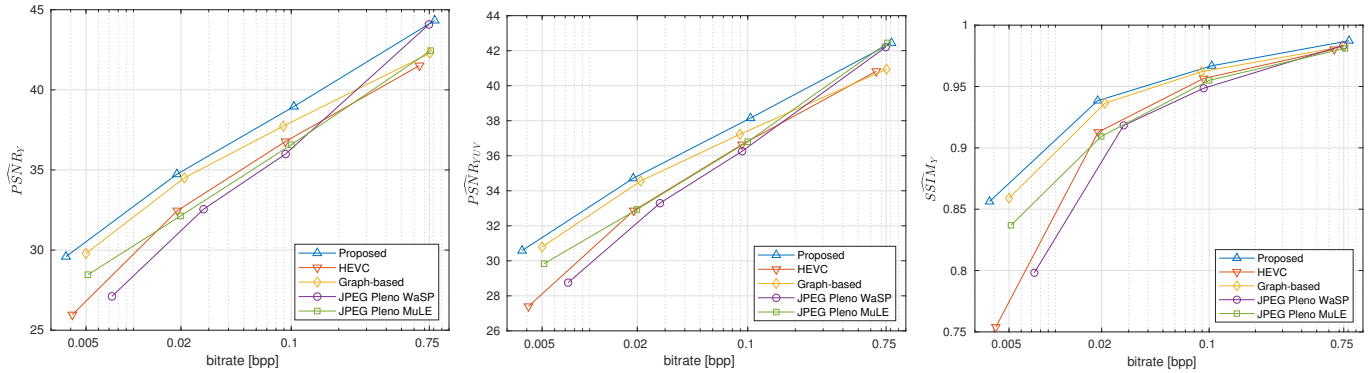
Results confirm that view estimation is a valid approach to reduce the information to be transmitted without compromising on the visual quality. In particular, both the graph-based method and our proposed architecture achieve the best results by encoding only a subset of views, relying on additional information to reconstruct the light field at the receiver side. Results demonstrate how this approach leads to a superior performance for light field compression.

V. Conclusions

In this paper we presented a new compression solution for light field images using view estimation and residual encoding. Our validating experiment shows that sensible gains can be achieved by using our solution against state-of-the-art compression algorithms. A MATLAB implementation of the proposed solution is available at `https://github.com/mmspg/light-field-translation-codec`.

Future work will focus on improving the performance of our proposed method for chroma channels. Moreover, the coding efficiency could be refined by implementing a multilevel architecture for view estimation.

[1] https://www.videolan.org/developers/x265.html



(a) $PSNR_Y$ (b) $PSNR_{YUV}$ (c) $SSIM_Y$

Fig. 4: Performance of the proposed encoding solution with respect to the selected anchors, here depicted for content *I09*.

TABLE III: Bjontegaard rate savings against the selected anchors HEVC/H.265, JPEG Pleno WaSP and MuLE, and the graph-based method of [3]. Negative values denote a decrease of bitrate for the same PSNR.

	HEVC/H.265		JPEG Pleno WaSP		JPEG Pleno MuLE		Graph-based	
	$\widehat{PSNR}_Y$	$\widehat{PSNR}_{YUV}$	$\widehat{PSNR}_Y$	$\widehat{PSNR}_{YUV}$	$\widehat{PSNR}_Y$	$\widehat{PSNR}_{YUV}$	$\widehat{PSNR}_Y$	$\widehat{PSNR}_{YUV}$
I01	−56.78%	−53.45%	**−64.92%**	**−60.37%**	−48.34%	−32.34%	−21.79%	−19.97%
I02	−59.22%	−56.52%	−55.24%	−47.39%	−40.61%	−20.85%	−24.36%	−22.42%
I04	**−63.12%**	**−59.67%**	−60.85%	−56.67%	−50.66%	−35.04%	−23.12%	−21.15%
I09	−54.73%	−49.69%	−58.89%	−52.06%	**−58.09%**	**−42.20%**	**−27.48%**	**−25.55%**
Average	−58.46%	−54.83%	−59.98%	−54.12%	−49.43%	−32.58%	−24.18%	−22.27%

TABLE IV: Bjontegaard dB gains against the selected anchors HEVC/H.265, JPEG Pleno WaSP and MuLE, and the graph-based method of [3]. Positive values denote an increase of PSNR for the same bitrate.

	HEVC/H.265		JPEG Pleno WaSP		JPEG Pleno MuLE		Graph-based	
	$\widehat{PSNR}_Y$	$\widehat{PSNR}_{YUV}$	$\widehat{PSNR}_Y$	$\widehat{PSNR}_{YUV}$	$\widehat{PSNR}_Y$	$\widehat{PSNR}_{YUV}$	$\widehat{PSNR}_Y$	$\widehat{PSNR}_{YUV}$
I01	2.62 dB	2.04 dB	**3.11 dB**	**2.23 dB**	1.86 dB	0.93 dB	0.74 dB	0.55 dB
I02	**2.72 dB**	**2.13 dB**	2.48 dB	1.65 dB	1.56 dB	0.59 dB	0.87 dB	0.64 dB
I04	2.64 dB	1.96 dB	2.55 dB	1.82 dB	1.92 dB	1.05 dB	0.70 dB	0.50 dB
I09	2.31 dB	1.69 dB	2.65 dB	1.79 dB	**2.32 dB**	**1.27 dB**	**0.93 dB**	**0.68 dB**
Average	2.57 dB	1.95 dB	2.70 dB	1.87 dB	1.92 dB	0.96 dB	0.81 dB	0.59 dB

REFERENCES

[1] X. Jiang, M. Le Pendu, R. A. Farrugia, and C. Guillemot, "Light field compression with homography-based low-rank approximation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1132–1145, 2017.

[2] S. Zhao and Z. Chen, "Light field image coding via linear approximation prior," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017.

[3] Viola, I., Maretic, H. P., Frossard, P. and Ebrahimi, T., "A graph learning approach for light field compression," 2018.

[4] P. Astola and I. Tabus, "Light Field Compression of HDCA Images Combining Linear Prediction and JPEG 2000," *EUSIPCO 2018*, 2018.

[5] ——, "Wasp: Hierarchical warping, merging, and sparse prediction for light field image compression," in *2018 7th European Workshop on Visual Information Processing (EUVIP)*. IEEE, 2018, pp. 1–6.

[6] ISO/IEC JTC 1/SC29/WG1 JPEG, "Verification Model Software Version 2.1 on JPEG Pleno Light Field Coding," Doc. N83034, Geneva, Switzerland, March 2019.

[7] M. Rizkallah, X. Su, T. Maugey, and C. Guillemot, "Graph-based transforms for predictive light field compression based on super-pixels," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP 2018*, 2018.

[8] X. Su, M. Rizkallah, T. Maugey, and C. Guillemot, "Rate-distortion optimized super-ray merging for light field compression," in *European Signal Processing Conference (EUSIPCO)*, 2018.

[9] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. da Silva, C. L. Pagliari, F. Pereira, and V. Testoni, "A 4D DCT-Based Lenslet Light Field Codec," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 435–439.

[10] Viola, I. and Ebrahimi, T., "Quality assessment of compression solutions for ICIP 2017 Grand Challenge on light field image coding," *2018 International Conference on Multimedia and Expo Workshops*, 2018.

[11] M. Řeřábek and T. Ebrahimi, "New light field image dataset," 2016.

[12] Dansereau, D. G., Pizarro, O., and Williams, S. B, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[13] ——, "Linear volumetric focus for light field cameras," *ACM Transactions on Graphics (TOG)*, 2015.

[14] ISO/IEC JTC 1/SC29/WG1 JPEG, "JPEG PLENO - Light Field Coding Common Test Conditions." 2018.